IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

VERTICAL COMPUTER SYSTEMS, INC.,	§	
	§	
Plaintiff,	§	
	§	
v.	§	CASE NO. 2:10-CV-490-JRG
	§	
INTERWOVEN, INC., et al.,	§	
	§	
Defendants.	§	
	§	
	§	
	§	

MEMORANDUM OPINION AND ORDER

Before the Court are Plaintiff Vertical Computer Systems, Inc.'s ("Plaintiff's" or "Vertical's") Opening Brief (Dkt. No. 130), the response of Defendants LG Electronics Mobilecomm USA, Inc., LG Electronics, Inc. (collectively, "LG"), Samsung Electronics Co., Ltd., and Samsung Electronics America, Inc. (collectively, "Samsung") (LG and Samsung are referred to collectively as "Defendants") (Dkt. No. 136), and Plaintiff's reply (Dkt. No. 138).

The Court held a claim construction hearing on September 4, 2013.

Table of Contents

I. BACKGROUND

Plaintiff brings suit alleging infringement of United States Patents No. 6,826,744 ("the '744 Patent") and 7,716,629 ("the '629 Patent") (collectively, "the patents-in-suit") (Dkt. No. 130, Exs. A & B). The patents-in-suit are both titled "System and Method for Generating Web Sites in an Arbitrary Object Framework." The '744 Patent issued on November 30, 2004, and bears a filing date of October 1, 1999. The '629 Patent issued on May 11, 2010, and is a continuation of the '744 Patent. In general, the patents-in-suit relate to creating computer software applications, such as for Internet web sites. The Abstract of the '744 Patent states:

> A system and method for generating computer applications in an arbitrary object framework. The method separates content, form, and function of the computer application so that each may be accessed or modified separately. The method includes creating arbitrary objects, managing the arbitrary objects throughout their life cycle in an object library, and deploying the arbitrary objects in a design framework for use in complex computer applications.

The Abstract of the '629 Patent states:

> A method and system for generating a computer application is disclosed. The computer application is generated on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application and a functionality of said computer application. Arbitrary objects are created with corresponding arbitrary names of various object types for generating said content of said computer application, said form of said computer application, and said functionality of said computer application. The arbitrary objects are managed in an object library. The arbitrary objects are deployed from said object library into a design framework to create said computer application.

The written descriptions of the patents-in-suit are substantially identical,[1] and the patents-in-suit have the same figures. For convenience, references to the specification and figures shall be to the '744 Patent only.

[1] Defendants argue:

> [Plaintiff's] representation that the differences between the '744 and '629 specifications are merely "typographical corrections" (Vertical Br. at 6) is wrong.

In *Vertical Computer Systems, Inc. v. Microsoft*, No. 2:07-CV-144 (E.D. Tex.) ("*Microsoft*"), the parties submitted claim construction briefing to the Court but settled prior to the claim construction hearing.

In *Interwoven, Inc. v. Vertical Computer Systems, Inc.*, the Northern District of California construed various disputed terms in the patents-in-suit, some of which are also disputed here. No. C 10-4645 RS, 2011 WL 6936186 (N.D. Cal. Dec. 30, 2011) (Seeborg, J.) ("*Interwoven*"). The relevant *Interwoven* constructions are set forth as to particular disputed terms, below.

After *Interwoven* construed the patents-in-suit, both patents-in-suit underwent *ex parte* reexamination at the United States Patent and Trademark Office ("PTO") pursuant to requests by Interwoven Inc. As to the '744 Patent, a reexamination certificate issued on November 16, 2012. (Dkt. No. 130, Ex. F.) As to the '629 Patent, a reexamination certificate issued on October 12, 2012. (*Id.*, Ex. I.)

The *Interwoven* court later revisited its construction of the term "arbitrary objects" while ruling on a motion for summary judgment. No. C 10-4645 RS, 2013 WL 3786633 (N.D. Cal. July 18, 2013) (Seeborg, J.) ("*Interwoven MSJ*").

II. LEGAL PRINCIPLES

It is understood that "[a] claim in a patent provides the metes and bounds of the right which the patent confers on the patentee to exclude others from making, using or selling the protected invention." *Burke, Inc. v. Bruno Indep. Living Aids, Inc.*, 183 F.3d 1334, 1340 (Fed.

> In [Plaintiff's] earlier case against Microsoft, Microsoft discovered that WebOS had been on sale more than a year before the filing date of the '744 patent. In an effort to avoid invalidity, Vertical subsequently amended the then-pending '629 specification from describing WebOS as an "embodiment of the present invention" to an "environment of the present invention." '629 patent at 5:6-7.

(Dkt. No. 136, at 2 n.1.)

Cir. 1999). Claim construction is clearly an issue of law for the court to decide. *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 970-71 (Fed. Cir. 1995) (en banc), *aff'd*, 517 U.S. 370 (1996).

To ascertain the meaning of claims, courts look to three primary sources: the claims, the specification, and the prosecution history. *Markman*, 52 F.3d at 979. The specification must contain a written description of the invention that enables one of ordinary skill in the art to make and use the invention. *Id.* A patent's claims must be read in view of the specification, of which they are a part. *Id.* For claim construction purposes, the description may act as a sort of dictionary, which explains the invention and may define terms used in the claims. *Id.* "One purpose for examining the specification is to determine if the patentee has limited the scope of the claims." *Watts v. XL Sys., Inc.*, 232 F.3d 877, 882 (Fed. Cir. 2000).

Nonetheless, it is the function of the claims, not the specification, to set forth the limits of the patentee's invention. Otherwise, there would be no need for claims. *SRI Int'l v. Matsushita Elec. Corp.*, 775 F.2d 1107, 1121 (Fed. Cir. 1985) (en banc). The patentee is free to be his own lexicographer, but any special definition given to a word must be clearly set forth in the specification. *Intellicall, Inc. v. Phonometrics, Inc.*, 952 F.2d 1384, 1388 (Fed. Cir. 1992). Although the specification may indicate that certain embodiments are preferred, particular embodiments appearing in the specification will not be read into the claims when the claim language is broader than the embodiments. *Electro Med. Sys., S.A. v. Cooper Life Sciences, Inc.*, 34 F.3d 1048, 1054 (Fed. Cir. 1994).

This Court's claim construction analysis is substantially guided by the Federal Circuit's decision in *Phillips v. AWH Corporation*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc). In *Phillips*, the court set forth several guideposts that courts should follow when construing claims. In

particular, the court reiterated that "the claims of a patent define the invention to which the patentee is entitled the right to exclude." 415 F.3d at 1312 (quoting *Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1115 (Fed. Cir. 2004)). To that end, the words used in a claim are generally given their ordinary and customary meaning. *Id.* The ordinary and customary meaning of a claim term "is the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention, i.e., as of the effective filing date of the patent application." *Id.* at 1313. This principle of patent law flows naturally from the recognition that inventors are usually persons who are skilled in the field of the invention and that patents are addressed to, and intended to be read by, others skilled in the particular art. *Id.*

Despite the importance of claim terms, *Phillips* made clear that "the person of ordinary skill in the art is deemed to read the claim term not only in the context of the particular claim in which the disputed term appears, but in the context of the entire patent, including the specification." *Id.* Although the claims themselves may provide guidance as to the meaning of particular terms, those terms are part of "a fully integrated written instrument." *Id.* at 1315 (quoting *Markman*, 52 F.3d at 978). Thus, the *Phillips* court emphasized the specification as being the primary basis for construing the claims. *Id.* at 1314-17. As the Supreme Court stated long ago, "in case of doubt or ambiguity it is proper in all cases to refer back to the descriptive portions of the specification to aid in solving the doubt or in ascertaining the true intent and meaning of the language employed in the claims." *Bates v. Coe*, 98 U.S. 31, 38 (1878). In addressing the role of the specification, the *Phillips* court quoted with approval its earlier observations from *Renishaw PLC v. Marposs Societa' per Azioni*, 158 F.3d 1243, 1250 (Fed. Cir. 1998):

> Ultimately, the interpretation to be given a term can only be determined and confirmed with a full understanding of what the inventors actually invented and

> intended to envelop with the claim. The construction that stays true to the claim language and most naturally aligns with the patent's description of the invention will be, in the end, the correct construction.

Phillips, 415 F.3d at 1316. Consequently, *Phillips* emphasized the important role the specification plays in the claim construction process.

The prosecution history also continues to play an important role in claim interpretation. Like the specification, the prosecution history helps to demonstrate how the inventor and the PTO understood the patent. *Id.* at 1317. Because the file history, however, "represents an ongoing negotiation between the PTO and the applicant," it may lack the clarity of the specification and thus be less useful in claim construction proceedings. *Id.* Nevertheless, the prosecution history is intrinsic evidence that is relevant to the determination of how the inventor understood the invention and whether the inventor limited the invention during prosecution by narrowing the scope of the claims. *Id.*; *see Microsoft Corp. v. Multi-Tech Sys., Inc.*, 357 F.3d 1340, 1350 (Fed. Cir. 2004) (noting that "a patentee's statements during prosecution, whether relied on by the examiner or not, are relevant to claim interpretation").

Phillips rejected any claim construction approach that sacrificed the intrinsic record in favor of extrinsic evidence, such as dictionary definitions or expert testimony. The *en banc* court condemned the suggestion made by *Texas Digital Systems, Inc. v. Telegenix, Inc.*, 308 F.3d 1193 (Fed. Cir. 2002), that a court should discern the ordinary meaning of the claim terms (through dictionaries or otherwise) before resorting to the specification for certain limited purposes. *Phillips*, 415 F.3d at 1319-24. According to *Phillips*, reliance on dictionary definitions at the expense of the specification had the effect of "focus[ing] the inquiry on the abstract meaning of words rather than on the meaning of claim terms within the context of the patent." *Id.* at 1321.

Phillips emphasized that the patent system is based on the proposition that the claims cover only the invented subject matter. *Id.*

Phillips does not preclude all uses of dictionaries in claim construction proceedings. Instead, the court assigned dictionaries a role subordinate to the intrinsic record. In doing so, the court emphasized that claim construction issues are not resolved by any magic formula. The court did not impose any particular sequence of steps for a court to follow when it considers disputed claim language. *Id.* at 1323-25. Rather, *Phillips* held that a court must attach the appropriate weight to the intrinsic sources offered in support of a proposed claim construction, bearing in mind the general rule that the claims measure the scope of the patent grant.

Indefiniteness is a "legal conclusion that is drawn from the court's performance of its duty as the construer of patent claims." *Exxon Research & Eng'g Co. v. United States*, 265 F.3d 1371, 1376 (Fed. Cir. 2001) (citation omitted). A finding of indefiniteness must overcome the statutory presumption of validity. *See* 35 U.S.C. § 282. That is, the "standard [for finding indefiniteness] is met where an accused infringer shows by clear and convincing evidence that a skilled artisan could not discern the boundaries of the claim based on the claim language, the specification, and the prosecution history, as well as her knowledge of the relevant art area." *Halliburton Energy Servs., Inc. v. M-I LLC*, 514 F.3d 1244, 1249-50 (Fed. Cir. 2008).

> In determining whether that standard is met, i.e., whether the claims at issue are sufficiently precise to permit a potential competitor to determine whether or not he is infringing, we have not held that a claim is indefinite merely because it poses a difficult issue of claim construction. We engage in claim construction every day, and cases frequently present close questions of claim construction on which expert witnesses, trial courts, and even the judges of this court may disagree. Under a broad concept of indefiniteness, all but the clearest claim construction issues could be regarded as giving rise to invalidating indefiniteness in the claims at issue. But we have not adopted that approach to the law of indefiniteness. We have not insisted that claims be plain on their face in order to avoid condemnation for indefiniteness; rather, what we have asked is that the claims be amenable to construction, however difficult that task may be. If a claim

> is insolubly ambiguous, and no narrowing construction can properly be adopted, we have held the claim indefinite. If the meaning of the claim is discernible, even though the task may be formidable and the conclusion may be one over which reasonable persons will disagree, we have held the claim sufficiently clear to avoid invalidity on indefiniteness grounds. . . . By finding claims indefinite only if reasonable efforts at claim construction prove futile, we accord respect to the statutory presumption of patent validity . . . and we protect the inventive contribution of patentees, even when the drafting of their patents has been less than ideal.

Exxon, 265 F.3d at 1375 (citations and internal quotation marks omitted).

In general, prior claim construction proceedings involving the same patents-in-suit are "entitled to reasoned deference under the broad principals of *stare decisis* and the goals articulated by the Supreme Court in *Markman*, even though *stare decisis* may not be applicable *per se*." *Maurice Mitchell Innovations, LP v. Intel Corp.*, No. 2:04-CV-450, 2006 WL 1751779, at *4 (E.D. Tex. June 21, 2006) (Davis, J.). The Court nonetheless conducts an independent evaluation during claim construction proceedings. *See, e.g., Texas Instruments, Inc. v. Linear Techs. Corp.*, 182 F. Supp. 2d 580, 589-90 (E.D. Tex. 2002); *Burns, Morris & Stewart Ltd. P'ship v. Masonite Int'l Corp.*, 401 F. Supp. 2d 692, 697 (E.D. Tex. 2005); *Negotiated Data Solutions, Inc. v. Apple, Inc.*, No. 2:11-CV-390, 2012 WL 6494240, at *5 (E.D. Tex. Dec. 13, 2012).

III. CONSTRUCTION OF DISPUTED TERMS

"In an effort to narrow the disputes presented to the Court, Defendants [have] withdraw[n] their request to construe 'object library' and 'design framework.'" (Dkt. No. 136, at 9 n.2.) Plaintiff had proposed that the Court construe those terms to have their plain and ordinary meaning. (Dkt. No. 127, Ex. A, at 6 & 7.) Because "object library" and "design framework" are no longer disputed by the parties, the Court does not address those terms.

A. "arbitrary object(s)" and Related Terms

"arbitrary object(s)" ('744 Patent, Claims 1, 16-19, 21-22, 24-25, 38-44, 46-47, 52 & 54-57; '629 Patent, Claims 1, 4-9, 11-13, 15-21, 24-29 & 31-32)	
Plaintiff's Proposed Construction	**Defendants' Proposed Construction**
"an object that can be created independently by individual preference, is interchangeable, and that may be, but need not be, accessed solely by name, the object being an entity that can have form, content, or functionality or any combination of form, content and functionality"	"objects that can be called by name with parameters and by name without parameters, that can be replaced by any arbitrary object of any different type, and that are not Java objects"
"that are interchangeable" and "each arbitrary object further being interchangeable with other arbitrary objects" ('744 Patent, Claims 56 & 57; '629 Patent, Claims 1, 13 & 21)	
Plaintiff's Proposed Construction	**Defendants' Proposed Construction**
Plain and ordinary meaning	"that can be replaced by any arbitrary object of any different type"
"that may be, but need not be, accessed solely by name" and "each arbitrary object being callable by name only" ('744 Patent, Claims 56 & 57; '629 Patent, Claims 1, 13 & 21)	
Plaintiff's Proposed Construction	**Defendants' Proposed Construction**
Plain and ordinary meaning	"that can be called by name with parameters and by name without parameters"

(Dkt. No. 127, Ex. A, at 2-4; *id.*, Ex. B, at 2-4.) As to the first of these disputed terms, Plaintiff proposes substantially the construction reached by *Interwoven*. *See* 2011 WL 6936186, at *12. During reexamination, the patentee amended the claims such that Claims 56 and 57 of the '744 Patent and Claims 1 and 13 of the '629 Patent now recite substantially the *Interwoven* construction for "arbitrary object(s)."

(1) The Parties' Positions

Plaintiff argues that the inventors coined the term "arbitrary object" and that *Interwoven* properly construed the term. (Dkt. No. 130, at 10.) Plaintiff argues that there is no support for excluding Java objects and that Defendants' proposal of "with parameters *and* by name" "makes the claim nonsensical and describes a technical impossibility." (*Id.*, at 11; *see id.*, at 13-14.) Plaintiff also argues that "[t]he specifications of the patents-in-suit not only do not make interchangeability of 'different types' mandatory, they do not make interchangeability mandatory." (*Id.*, at 12.) Plaintiff further argues claim differentiation as to Claim 29 of the '629 Patent, which recites "executable instructions for swapping an arbitrary object of one type with an arbitrary object of another type." (*Id.*, at 13.) As to the constituent parts of the *Interwoven* construction that now appear in claims that the patentee amended during reexamination, Plaintiff argues that "[t]he patents-in-suit describe these separate features or characteristics of arbitrary objects in an unambiguous manner that does not require any clarification." (*Id.*, at 15.)

Defendants respond that "the retrieval of objects by name only was known years before the patents-in-suit were filed." (Dkt. No. 136, at 2.) Defendants argue that "[t]he patents-in-suit, however, never actually explain how an arbitrary object can be called by name only." (*Id.*, at 3.) Defendants also argue that during prosecution, the patentee repeatedly distinguished prior art "classic objects" as requiring the passing of parameters. (*Id.*, at 10.) Defendants submit that the claim language, which recites "'that may be, but need not be, accessed solely by name[,]' means that arbitrary objects may be 'accessed solely by name' as well as by name with parameters." (*Id.*, at 11 (emphasis Defendants').) As to Plaintiff's concern regarding Defendants' use of the word "and," Defendants respond:

> Defendants' construction requires that an arbitrary object be capable of being called both by name with parameters and by name without parameters. It does

> not require, as [Plaintiff] suggests, that an arbitrary object be called both ways at the same time.

(*Id.*) Defendants also submit that in *Interwoven*, Plaintiff "insist[ed] that what distinguishes its invention from the prior art is that . . . with the new invention, a programmer can choose either to retrieve the object by name or pass parameters as per the traditional model." (*Id.*, at 12 (quoting *Interwoven*, 2011 WL 6936186, at *11).)

As to interchangeability, Defendants cite the specification, prosecution history, and the analysis in *Interwoven*. (Dkt. No. 136, at 13-15.) Defendants also argue that *Interwoven* rejected Plaintiff's claim differentiation argument. (*Id.*, at 15-16.) Defendants conclude that "[a]s in *Interwoven*, Defendants' construction describes the characteristic of interchangeability, not the act of doing so." (*Id.*, at 16.)

As to Java objects, Defendants respond that "[b]y taking the position that Java objects were not 'arbitrary objects' to overcome the Leshem reference [(United States Patent Application Publication No. 2002/0147805)], [the patentee] disclaimed Java objects." (Dkt. No. 136, at 17-18.)

Plaintiff replies by reiterating its claim differentiation argument as to interchangeability and by addressing the related findings by the Northern District of California in *Interwoven MSJ*, in particular the court's rejection of a "different types" limitation. (Dkt. No. 138, at 2-3.) Plaintiff also argues that "the term Java objects is completely ambiguous" and that in the prosecution history cited by Defendants, the patentee "also refer[red] to several other 'classic' objects such as CGI, and then merely state[d] a fact: a 'classic' Java, CGI or other 'classic' object could not be called by name only, and was not interchangeable." (*Id.*, at 4 & 5.) Plaintiff concludes that "[b]ecause 'classic' Java objects, C++ objects and others which are not callable by name only and cannot be interchanged can be used in combination to create totally new

entities such as arbitrary object frameworks and arbitrary objects that are callable by name only and are interchangeable, the exclusion of Java objects in the definition of arbitrary objects makes absolutely no sense." (*Id.*, at 6.) Finally, Plaintiff submits that "[w]ith respect to calling objects with parameters or without parameters, the defendants do not provide any cogent basis for using the conjunctive." (*Id.*)

At the September 4, 2013 hearing, Plaintiff reiterated that Defendants' proposal would requiring calling an arbitrary object both with and without parameters, simultaneously, which Plaintiff explained is impossible. Defendants responded that they propose an arbitrary object is callable by name with parameters and is callable by name without parameters, not that an arbitrary object can be called with parameters and without parameters at the same time. Defendants nonetheless expressed concern that Plaintiff's proposed construction could be interpreted to encompass classic objects, which must be called by name with parameters.

As to interchangeability, Defendants stressed they are arguing that an arbitrary object must be *capable* of being swapped with another arbitrary object of a different type, not that an arbitrary object *must* be so swapped. In other words, Defendants argued, an arbitrary object must be swappable with other arbitrary objects of the same type or of a different type. Defendants also argued that Plaintiff's claim differentiation argument should be rejected because whereas the dependent claims recite actual swapping with another arbitrary object of a different type, the independent claims merely recite the capability of swapping regardless of type.

Finally, as to Java, Defendants urged that although the specification suggests that any programming language can be used, the patentee specifically disclaimed Java during prosecution. Defendants also submitted that claims can be construed with a view toward resolving infringement issues. *See Typhoon Touch Techs., Inc. v. Dell, Inc.*, 659 F.3d 1376,

1383 (Fed. Cir. 2011) ("It is not inappropriate for a court to consider the accused devices when construing claim terms, for the purpose of 'claim construction' is to resolve issues of infringement."). Plaintiff responded by noting that the specification never refers to Java and by arguing that "Java" is a generic label that should not be injected into the claim.

(2) Analysis

(a) Parameters

Interwoven highlighted the ability of "arbitrary objects" to operate both when passed parameters and when called by name without parameters:

> With this innovation, programmers may access an arbitrary object even if they do not know the exact parameters of the object; it is in this instance that an arbitrary name would be used. If they do know the parameters, however, programmers can decide to use either the arbitrary name or pass parameters as they did in the traditional model. Nothing in the patent language mandates the restriction of arbitrary objects to program pieces only invoked by their corresponding names.

Interwoven, 2011 WL 6936186, at *12.

The specification states that an "arbitrary object can be accessed by an arbitrary object name." ('744 Patent at 4:39-40; *see also id.* at 5:42-46 ("Many functions are stored within an object library on an arbitrary object framework such that those functions can be accessed by name arbitrarily. This is in contrast to a traditional model where the function must be explicitly invoked with all its parameters included."); *id.* at 5:54-55 ("All that is needed is the name of the function in order to access the function.").) The specification further emphasizes that "[a] critical distinction between the present invention and previous object oriented development systems is the need to know how a function can be called and what to expect it to return, rather than just knowing the function's name." (*Id.* at 5:62-65.)

During prosecution, the patentee distinguished arbitrary objects from so-called "classic objects," which "require the passing of parameters and knowing what to expect in return . . . ."

(Dkt. No. 136, Ex. M, 5/31/2012 Amendment ('744 Patent Reexamination), at 13; *see also id.*, Ex. S, 8/21/2012 Amendment ('629 Patent Reexamination), at 12; *id.*, Ex. O, 8/21/2012 Amendment ('744 Patent Reexamination), at 13 ("If parameters are required to execute such objects, then the correct parameters must be passed when the object is called and hence cannot be called by name only.") (emphasis omitted).)

By contrast, "according to principles of the [patents in suit], arbitrary objects may be called by name only . . . ." (*Id.*, Ex. O, 8/21/2012 Amendment ('744 Patent Reexamination), at 14; *see also id.*, at 12 ("Among other attributes, the invention herein, also referred to as arbitrary objects, may be called by name only, that is, with or without parameters, because there is no need to explicitly pass parameters (see, *e.g.*, col. 5, lines 43-65) . . . ."); *id.*, Ex. M, 5/31/2012 Amendment ('744 Patent Reexamination), at 13 & 14; *id.*, Ex. S, 8/21/2012 Amendment ('629 Patent Reexamination), at 11 & 12; *id.*, Ex. K, 12/30/2009 Amendment and Response ('629 Patent), at 16 ("In clear contrast [to the prior art of record], it is sufficient with Applicant's arbitrary objects to just know a function's name to call it . . . ."); *id.*, at 14 ("[A]rbitrary objects can be called by file name only, because there is no need to explicitly pass parameters . . . .").)

The specification and the prosecution history thus demonstrate that, while not necessary that an "arbitrary object" be called by name only in every instance, i.e., it may sometimes be called by name *with* parameters, the claims require that an arbitrary object has the capacity to be called by name only, without passing parameters.

(b) Interchangeability

Interwoven found that despite the appearance of permissive language, interchangeability is a limitation:

> Vertical's assertion that the permissive expressions such as "can be swapped," and "can be easily replaced" means that the patentee did not mandate that arbitrary objects be interchangeable, is inaccurate. Just because terms such as "can" and "may" appear, does not automatically negate an imposed limitation. Rather the patentee's assertion that arbitrary objects "can be easily replaced," means that they must have this attribute; the patentee excluded all objects that cannot be replaced from the definition. Arbitrary objects, therefore, must be interchangeable as per Interwoven's proposed construction.

Interwoven, 2011 WL 6936186, at *10.

> [W]hile not necessary that an arbitrary object be replaced with an object of a different type in every instance, the claimed language requires that a product falling within the reach of the patent have the *capacity*, but is not required, to exchange one arbitrary object for an object of another type.

Interwoven MSJ, slip op. at 7.

The specification likewise states that "[t]he present invention allows different object types to be interchangeable." '744 Patent at 6:13-14. "One arbitrary object can be easily replaced with another arbitrary object of another type." *Id.* at 4:41-42; *see also id.* at 3:58-61 ("Because the object pointers are not tied in any way to the functionality of the object, an object of one type can be easily replaced with another object of another type.").

During prosecution, the patentee distinguished prior art references by again reiterating that "arbitrary objects of any type can be readily replaced with another arbitrary object of another type, *i.e.*, arbitrary objects are *interchangeable* with each other, *regardless of type* . . . ." (Dkt. No. 126, Ex. K, 12/30/2009 Amendment and Response ('629 Patent), at 14 (emphasis modified); *see also id.*, at 15 ("In clear contrast to arbitrary objects, prior art object-oriented systems, such as Leshem, utilize 'classic' objects . . . which are <u>not interchangeable</u> with other objects of different types . . . .") (underlining in original; italics omitted); *id.*, Ex. M, 5/31/2012 Amendment ('744 Patent Reexamination), at 13 ("[A]rbitrary objects of any type can also be readily replaced with another arbitrary object of <u>another type</u>, *i.e.*, arbitrary objects are

interchangeable, or swappable, with each other, regardless of type."); *id.*, Ex. O, 8/21/2012 Amendment ('744 Patent Reexamination), at 14 (same); *id.*, Ex. S, 8/21/2012 Amendment ('629 Patent Reexamination), at 12-13 (same).)

> In clear contrast to the interchangeability or swappability of arbitrary objects, because prior art object-oriented systems, such as Borland [("Borland® Delphi 3 for Windows 95 & Windows NT, User's Guide" (1997))] and Morgan utilize "classic" objects which cannot be called by name only, and require the passing of parameters and knowing what to expect in return, objects are not interchangeable or swappable with other objects of different types. Rather, Borland replaces one object type in a project with another object of the same type. Form objects, function objects, and content objects are all "swapped" (*i.e.*, added or removed) to and from a project, but that is not the same as, *e.g.*, swapping a form object for a content object or a function object.

(Dkt. No. 136, Ex. M, 5/31/2012 Amendment ('744 Patent Reexamination), at 13 (underlining in original; italics modified.)

As for claim differentiation, Claim 29 of the '629 Patent recites: "The system of claim 21, further comprising executable instructions for swapping an arbitrary object of one type with an arbitrary object of another type." Claim 44 of the '744 Patent is similar. These dependent claims are further evidence that an arbitrary object has the ability to be swapped with an arbitrary object of another, different type *or* with an arbitrary object of the *same* type. *See Wenger Mfg., Inc. v. Coating Mach. Sys., Inc.*, 239 F.3d 1225, 1233 (Fed. Cir. 2001) ("Claim differentiation, while often argued to be controlling when it does not apply, is clearly applicable when there is a dispute over whether a limitation found in a dependent claim should be read into an independent claim, and that limitation is the only meaningful difference between the two claims.").

The intrinsic evidence thus establishes that an arbitrary object is interchangeable with an arbitrary object of the same type or an arbitrary object of a different type.

(c) Java

Defendants seek to introduce a negative limitation that the "arbitrary objects" cannot be "Java" objects. Java is a so-called "object-oriented" programming language. *See Interwoven*, 2011 WL 6936186, at *1. This issue of excluding Java objects does not appear to have been raised in the *Interwoven* case.

As a general matter, a negative limitation can be included as part of the construction of a disputed term if appropriate. *See, e.g., N. Am. Container, Inc. v. Plastipak Packaging, Inc.*, 415 F.3d 1335, 1343-45 (Fed. Cir. 2005) (affirming construction of "generally convex" to require "no concave points," because "[t]o overcome an obviousness rejection, the applicant distinguished his invention from the [prior art] on the basis of the latter disclosing inner walls that are 'slightly concave.' The inescapable consequence of such an argument is that the scope of applicant's claims cannot cover inner walls that are 'slightly concave.'").

Negative limitations generally require an "anchor" in the claim language or in the specification. *Omega Eng'g, Inc. v. Raytek Corp.*, 334 F.3d 1314, 1322 (Fed. Cir. 2003) (finding that district court's "additional negative limitation finds no anchor in the explicit claim language"); *see Linear Tech. Corp. v. Int'l Trade Comm'n*, 566 F.3d 1049, 1060 (Fed. Cir. 2009) (finding that "there is no basis in the patent specification for adding the negative limitation").

During prosecution, the patentee discussed Java:

> Leshem, paragraph [0121], makes reference to several "content or service" types, including Java, "other applications" and CGI. These objects are what Applicant would refer to as "function" objects. These objects (e.g., Java, CGI, etc.) cannot be called by name only. By way of example, a java program that requires parameters will not function properly if called by name without the parameters. Further, *a Java object cannot be replaced with a CGI object without implementing additional changes*. Thus, in clear contrast to Applicant's arbitrary objects, the objects disclosed by Leshem cannot be called by name only or interchanged with other objects of other types.

(Dkt. No. 136, Ex. K, 12/30/2009 Amendment ('629 Patent), at 15-16 (underlining in original; italics modified).)

The Court has already found, above, that interchangeability regardless of type is a required feature of the claimed "arbitrary objects." Assuming *arguendo* that Java objects are not interchangeable with objects of different types, introducing a negative limitation regarding Java is unnecessary and would tend to confuse rather than clarify the scope of the claims. *See PACT XPP Techs., AG v. Xilinx, Inc.*, No. 2:07-CV-563, 2011 WL 2469909, at *18 (E.D. Tex. June 17, 2011) (Everingham, J.) (finding that the proposed "negative limitation is not required and will only confuse the jury").

The Court therefore hereby expressly rejects Defendants' proposal of the phrase "that are not Java objects."

(d) Construction

As discussed above, "arbitrary objects" can be called by name with parameters or by name without parameters. To be clear, while it is not necessary that an "arbitrary object" be called by name only in every instance, i.e., it may sometimes be called by name with parameters, the claims require that an arbitrary object has the capacity to be called by name only without passing parameters.

As also discussed above, "arbitrary objects" are interchangeable with other arbitrary objects of the same type or of a different type.

Further, the Court expressly rejects Defendants' proposal to introduce a negative limitation excluding Java objects.

Finally, Plaintiff's proposal of "or any combination of form, content and functionality" as to the term "arbitrary object(s)" should be adopted for the reasons set forth as to the "that separates . . ." and "the object being . . ." terms, discussed in subsection C., below.

The Court therefore hereby construes the disputed terms as set forth in the chart below:

Term	Construction
"arbitrary object(s)"	**"object that can be created independently, that is interchangeable with other objects of the same type or of a different type, and that can be accessed by name with parameters or by name without parameters, the object being an entity that can have form, content, or functionality or any combination of form, content, and functionality"**
"that are interchangeable"	**"that are interchangeable with other objects of the same type or of a different type"**
"each arbitrary object further being interchangeable with other arbitrary objects"	**"each arbitrary object further being interchangeable with other objects of the same type or of a different type"**
"that may be, but need not be, accessed solely by name"	**"that can be accessed by name with parameters or by name without parameters"**
"each arbitrary object being callable by name only"	**"each arbitrary object can be called by name with parameters or by name without parameters"**

B. "created independently by individual preference"

Plaintiff's Proposed Construction	**Defendants' Proposed Construction**
Plain and ordinary meaning	Indefinite

(Dkt. No. 127, Ex. A, at 8; *id.*, Ex. B, at 8.) This disputed term appears in Claims 56 and 57 of the '744 Patent and Claim 1 of the '629 Patent.

(1) The Parties' Positions

Plaintiff argues that this term requires no construction because *Interwoven* included this phrase in its construction of "arbitrary objects." (Dkt. No. 130, at 14.) Plaintiff also argues that Defendants cannot meet the high burden for establishing indefiniteness. (*Id.*, at 16-17.)

Defendants respond that "the record does not define how the arbitrary objects are created independently and whose preference is used to create the arbitrary objects." (Dkt. No. 136, at 27.) Defendants also note that "individual preference" was proposed by both sides in *Interwoven*, so *Interwoven* did not substantively address that phrase. (*Id.*) Defendants conclude that "[w]ithout knowing what arbitrary objects are created independently from and whose preferences the creation is based on, this term is fatally ambiguous." (*Id.*, at 29.)

Plaintiff replies that in *Microsoft*, Microsoft and its counsel included the phrase "desired by the developer" in their proposed construction for the term "arbitrary object," and Plaintiff notes that Defendants in the present case are represented by some of the same counsel. (Dkt. No. 138, at 7 (citing Ex. O, *Vertical Computer Systems Inc. v. Microsoft Corporation*, No. 2:07-CV-144 (E.D. Tex.), 4/23/2008 Joint Claim Construction and Prehearing Statement, Ex. B, Microsoft's Claim Chart for U.S. Patent No. 6,826,744, at 10).)

(2) Analysis

General legal principles regarding indefiniteness are set forth in Section II., above. Claim 1 of the '629 Patent is representative and recites (as amended by the reexamination certificate; emphasis added):

> 1. A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:
>
> a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application each arbitrary object being separate from each other arbitrary object, said arbitrary objects being objects that can be *created independently by individual preference*, that are interchangeable, and that may be,

> but need not be, accessed solely by name, the object being an entity that can have form, content, or functionality or any combination of form, content and functionality;
>
> a second set of executable instructions for managing said arbitrary objects in an arbitrary object library; and
>
> a third set of executable instructions for deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.

On one hand, a term can render a claim indefinite if one of ordinary skill in the art would not be able to discern the scope of the claim. *See Honeywell Int'l Inc. v. Int'l Trade Comm'n*, 341 F.3d 1332, 1340 (Fed. Cir. 2003) (finding a claim term indefinite because "the claims, the written description, and the prosecution history fail to give us, as the interpreter of the claim term, any guidance as to what one of ordinary skill in the art would interpret the claim to require"); *see also Halliburton*, 514 F.3d at 1255-56 (holding that term "fragile gel" rendered claim indefinite because "the fluid is defined by what it does rather than what it is") (citation and internal quotation marks omitted).

On the other hand, the Summary of the Invention provides context by explaining the desirability of independent management of form, content, and function:

> A traditional approach such as that illustrated in FIG. 1, may create unwanted bottlenecks in the production process. Each upstream revision, such as a change of content 10 or design 12, forces a repetition of the entire process.
>
> * * *
>
> Therefore a need exists for a method of generating complex software applications that reduces or eliminates production delays and the workload for programmers due to changes in content and/or form. This method should separate form, content and function so that each area can be independently changed.

'744 Patent at 1:37-40 & 1:66-2:15.

Generally, "claims are interpreted with an eye toward giving effect to all terms in the claim." *Bicon, Inc. v. Straumann Co.*, 441 F.3d 945, 950 (Fed. Cir. 2006). Nonetheless,

"surplusage may exist in some claims." *Decisioning.com, Inc. v. Federated Dep't Stores, Inc.*, 527 F.3d 1300, 1312 n.6 (Fed. Cir. 2008); *accord ERBE Elektromedizin GmbH v. Canady Tech. LLC*, 629 F.3d 1278, 1286 (Fed. Cir. 2010).

At the September 4, 2013 hearing, Defendants referred to page 8 of their responsive brief citing reexamination prosecution history in which the patentee specifically relied upon the "by individual preference" language in order to overcome prior art. (*See* Dkt. No. 136, Ex. O, 8/21/2012 Amendment ('744 Patent Reexamination), at 8-9; *see also id.*, Ex. P, 9/13/2012 Advisory Action ('744 Patent Reexamination); *id.*, Ex. Q, 10/4/2012 Amendment ('744 Patent Reexamination), at 2, 4 & 12; *id.*, Ex. S, 8/21/2012 Amendment ('629 Patent Reexamination), at 2, 4 & 10-13.) As Plaintiff represented in response at the hearing, no such reliance is evident in this cited prosecution history or any other prosecution history that has been presented to the Court.

On balance, the phrase "by individual preference" is surplusage that merely emphasizes that arbitrary objects for content, form, and functionality can be created independently, in any manner "desired." ('744 Patent at 3:42-43 & 4:20-21 ("Arbitrary objects may include any combination of application logic and data *desired by a developer*.") (emphasis added).) This is analogous to a non-limiting "whereby" clause "that merely states the result of the limitations in the claim" and that "adds nothing to the patentability or substance of the claim." *Tex. Instruments Inc. v. U.S. Int'l Trade Comm'n*, 988 F.2d 1165, 1172 (Fed. Cir. 1993); *accord Lockheed Martin Corp. v. Space Sys./Loral, Inc.*, 324 F.3d 1308, 1319 (Fed. Cir. 2003); *Plant Equipment, Inc. v. Intrado, Inc.*, No. 2:09-CV-395, 2012 WL 1468594, at *8 n.7 (E.D. Tex. Apr. 27, 2012) ("[T]he phrase 'so as to maintain privacy' indicates a result that follows. The phrase is

similar to a 'whereby' clause that merely states the results of the limitations in the claim and adds nothing to the substance of the claim.") (citing *Lockheed* and *Texas Instruments*).

Because the "by individual preference" language is not limiting, Defendants have failed to "show[] by clear and convincing evidence that a skilled artisan could not discern the boundaries of the claim based on the claim language, the specification, and the prosecution history, as well as her knowledge of the relevant art area." *Halliburton*, 514 F.3d at 1249-50. Defendants' indefiniteness argument is therefore hereby expressly rejected.

No other disputes between the parties being apparent, the Court hereby construes **"created independently by individual preference"** to have its **plain meaning**, with the further clarification that "by individual preference" is not a claim limitation.

C. "that separates a content of said computer application, a form of said computer application and a functionality of said computer application" and "the object being an entity that can have form, content, or functionality or any combination of form, content and functionality"

"that separates a content of said computer application, a form of said computer application and a functionality of said computer application" ('744 Patent, Claims 1, 54 & 56; '629 Patent, Claims 1 and 21)	
Plaintiff's Proposed Construction	**Defendants' Proposed Construction**
"the ability independently to modify and access Content, Form, and Function Objects"	"wherein the content, form, and functionality of a computer application are defined by content-only, form-only, and functionality-only arbitrary objects, respectively"
"the object being an entity that can have form, content, or functionality or any combination of form, content and functionality" ('744 Patent, Claims 56 & 57; '629 Patent, Claims 1 & 13)	
Plaintiff's Proposed Construction	**Defendants' Proposed Construction**
Plain meaning	Indefinite in view of the "that separates a content of said computer application, a form of said computer application and a functionality of said computer application" limitation

(Dkt. No. 127 at 4 & 5; *id.*, Ex. B, at 5). For the first of these disputed terms, Plaintiff proposes the construction reached by *Interwoven*. *See* 2011 WL 6936186, at *16.

(1) The Parties' Positions

(a) "that separates . . ."

Plaintiff argues that "[n]either the patents-in-suit nor their prosecution histories require that the method or system perform this separation all the time." (Dkt. No. 130, at 19.) "Alternatively, [Plaintiff] asserts that the subject phrase[, which appears in the preambles,] is not even a limitation of the claims that requires construction" because "[t]he body of the claims

includes the creation of arbitrary objects which allows the separation, and, thus, the preamble does not add anything which the body does not include." (*Id.*)

Defendants respond that the claim language itself distinguishes between content objects, form objects, and function objects. (Dkt. No. 136, at 19.) Defendants also urge that "the 'that separates' phrase recites 'the essence of the invention' and 'is necessary to give life, meaning, and vitality to the claim.'" (*Id.*, at 21-22 (quoting *IP Innovation, LLC v. Google, Inc.*, No. 2:07-CV-503, 2010 WL 104547, at *5 (E.D. Tex. Jan. 7, 2010)).) Defendants further submit that *Interwoven* found the preambles limiting in this regard, that the patentee relied upon the disputed preamble term during prosecution to distinguish prior art, and that the disputed preamble term provides antecedent basis, for example as to "said content" and "said functionality" in the body of Claim 1 of the '744 Patent. (Dkt. No. 136, at 22.)

Plaintiff replies that Defendants' proposal "improperly focuses on objects and what they may include," "does not focus on a framework – the subject that this phrase modifies," and "uses improper absolutes such as 'content-only' which the record simply does not support or even suggest." (Dkt. No. 138, at 8.)

(b) "the object being . . ."

Plaintiff argues that this term requires no construction because *Interwoven* included this phrase in its construction of "arbitrary objects." (Dkt. No. 130, at 14.) Plaintiff also argues that Defendants cannot meet the high burden for establishing indefiniteness. (*Id.*, at 16-17.)

Defendants respond that this disputed term allows for a "combination" even though the patentee "distinguished several prior art references [during prosecution] on the ground that the 'that separates' phrase prohibits arbitrary objects from including any combination of content, form and functionality of the computer application." (Dkt. No. 136, at 23-24.) Defendants

conclude that "[i]n a situation where – as here – two terms in the same claim conflict, the claim is invalid as indefinite." (*Id.*, at 24.)

Plaintiff replies "[i]t is the arbitrary object framework that separates, while arbitrary objects may have any combination of application logic and data. This term simply is not indefinite." (Dkt. No. 138, at 8.)

At the September 4, 2013 hearing, Plaintiff reiterated that separation of form, content, and functionality is not a feature of the arbitrary objects alone. Instead, Plaintiff explained, the framework, in combination with the arbitrary objects, accomplishes separation. Defendants responded that nothing in the specification explains how the framework would separate content, form, and functionality. Defendants concluded that the separation ability of the claimed systems and methods is due to the use of arbitrary objects that are form-only, content-only, and functionality-only. Plaintiff replied that the prosecution history cited by Defendants relates to separation (of form, content, and functionality) that occurs at the program level or system level, not at the object level.

(2) Analysis

As a threshold matter, the parties dispute whether the "that separates . . ." term is a limitation of the claims.

> In general, a preamble limits the invention if it recites essential structure or steps, or if it is "necessary to give life, meaning, and vitality" to the claim. *Pitney Bowes*[, *Inc. v. Hewlett-Packard Co.*], 182 F.3d [1298,] 1305 [(Fed. Cir. 1999)]. Conversely, a preamble is not limiting "where a patentee defines a structurally complete invention in the claim body and uses the preamble only to state a purpose or intended use for the invention." *Rowe v. Dror*, 112 F.3d 473, 478, 42 USPQ2d 1550, 1553 (Fed. Cir. 1997).

Catalina Mktg. Int'l, Inc. v. Coolsavings.com, Inc., 289 F.3d 801, 808 (Fed. Cir. 2002); *see, e.g.*, *In re Cruciferous Sprout Litig.*, 301 F.3d 1343, 1348 (Fed. Cir. 2002) (finding the preamble

limiting because of "clear reliance by the patentee on the preamble to persuade the Patent Office that the claimed invention is not anticipated by the prior art"); *Eaton Corp. v. Rockwell Int'l Corp.*, 323 F.3d 1332, 1339 (Fed. Cir. 2003) ("When limitations in the body of the claim rely upon and derive antecedent basis from the preamble, then the preamble may act as a necessary component of the claimed invention.").

Because the term "that separates a content of said computer application, a form of said computer application[,] and a functionality of said computer application" provides antecedent basis for "said content," "said form," and "said functionality" in the body of the claims, the "that separates . . ." term is a limitation of the claims in which it appears. *See id.*

As to the proper construction, on one hand the specification suggests that "combinations" may be included in an arbitrary object if desired:

> Arbitrary objects may include any combination of application logic and data desired by a developer. Arbitrary objects can include text file pointers, binary file pointers, compiled executables, scripts, data base queries, shell commands, remote procedure calls, global variables, and local variables.

('744 Patent at 3:42-46 & 4:20-25.)

On the other hand, the Background of the Invention and the Summary of the Invention explain advantages of separating content, form, and functionality:

> Three processes used to create complex software applications such as web sites are *form, function, and content*. Form includes graphic designs, user interfaces, and graphical representations created by a designer or a group of designers. Function includes logical functionality, which can be software code created by a programmer or group of programmers. Form includes informative content. Informative content can include written, recorded, or illustrated documentation, such as photographs, illustrations, product marketing material, and news articles. Content can be created by writers, photographers, artists, reporters, or editors.
>
> * * *
>
> More specifically, the present invention provides a method for generating software applications in an arbitrary object framework. *The method of the present*

> *invention separates content, form, and function of the computer application so that each may be accessed or modified independently*. . . .
>
> The *present invention* provides an important technical advantage in that *content, form, and function are separated from each other in the generation of the software application*. Therefore, changes in design or content do not require the intervention of a programmer. This advantage decreases the time needed to change various aspects of the software application. Consequently, cost is reduced and versatility is increased.

('744 Patent at 1:11-22 & 2:9-26 (emphasis added).) The specification likewise explains:

> The present invention provides a system and method for using a hierarchical, arbitrary object framework for generating software applications. *The method separates content, form, and function of the software application so that each can be accessed or modified independently.* The method of this invention includes creating arbitrary objects, managing the arbitrary objects in an object library, and deploying the arbitrary objects in a design framework for use in computer applications.
>
> * * * *
>
> *The hierarchical framework separates content 10, form 12, and functionality 14 to generate product 16.* Product 16 may be a computer software application such as a web site. Since content 10, design 12, and functionality 14 are separate entities independent of each other, modification in one group does not require corresponding modifications in another group. Each group can contribute to product 16 directly.

(*Id.* at 3:14-22 & 3:35-39 (emphasis added).)

During original prosecution of the '744 Patent, the patentee emphasized the separation of content, form, and functionality:

> Johnson [(United States Patent No. 6,052,670)] classes include[] two or more of "a content . . . a form . . . and a functionality of the computer application" and no set of Johnson classes separates "a content . . . a form . . . and a functionality of the computer application."
>
> * * *
>
> [T]he Johnson catalog class includes both form and content of the computer application disclosed in Johnson.

(Dkt. No. 136, Ex. B, 1/21/2003 Amendment and Response ('744 Patent), at 4-5.)

Moreover, *separating form, content, and function using arbitrary objects achieves many advantages in accordance with the claimed invention.* One such advantage is that when content, form, and functionality are genuinely separated in the generation of a software application, changes in one do not affect the other.

* * *

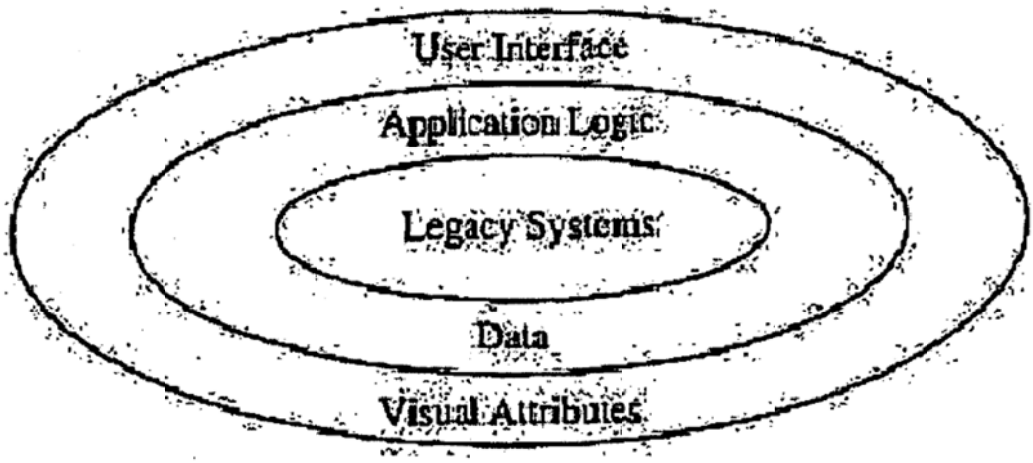
[T]he Lewandowski reference[2] does NOT, in fact, teach creating arbitrary objects for generating separate content, form, and functionality of a computer program. Respectfully, the Examiner misinterprets Figure 10 on page 22 of the Lewandowski reference to allege that a business logic object ("BLO") is used to generate content, a presentation object is used to represent form, and a business process object ("BPO") is used to represent functionality. As described on page 22 of the Lewandowski reference, the BLO defines how the object reacts to certain events, which clearly is a functional operation, not a content characteristic. Although the BLO is described as storing some business data, the BLO mixes the storing of business data with this functional component. Thus, the BLO of the Lewandowski reference still fails to resolve the problems associated with mixing content and functionality in a program.

* * *

Lewandowski also states that the three objects are "managed by one object . . . ." As described above, *one advantage of the claimed method is that the three object types operate separately from each other*, and therefore, changes on one object type would not affect the other object types. In contrast, given that the objects in the Lewandowski reference are managed by one object, if the managing object changes, then it will likely be necessary to change the other three objects, and vice versa.

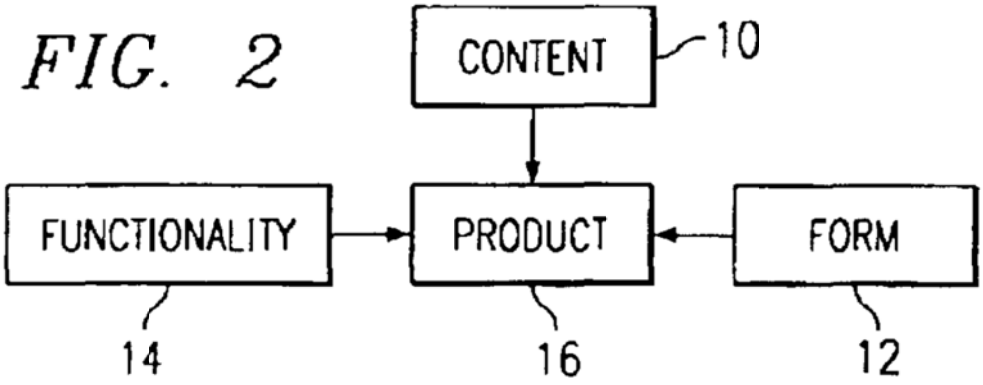
The diagram illustrating Lewandowski's application layers in Figure 11 of page 23, is particularly illustrative of Lewandowski's failure to separate content and functionality using objects. See below:

[2] The "Lewandowski" reference is cited on the cover page of the '744 Patent as: "Lewandowski, Framework for Component-Based Client/Server Computing, Mar. 1998, ACM, pp. 3-27."



The application logic layer, or tier as Lewandowski calls it, includes both application logic and data implementation. This is clearly contrary to, and disadvantageous when compared with, the claimed method of the present application. Changes in the format of the data in the Lewandowski reference, for example, can clearly affect the functioning of the application logic. Thus, it is abundantly clear that the Lewandowski reference *has not truly separated content, form, and functionality, using arbitrary objects* in a computer program, *as required by the claimed invention*.

However, Figure 2 of the present application illustrates that the interaction of the *content, form, and functionality objects* are clearly distinct from the objects utilized in the Lewandowski reference.



Thus, as clarified in Figure 2, there is *no overlap* or sharing of layers *with the claimed invention of the present application*.

(Dkt. No. 136, Ex. D, 10/2/2003 Response ('744 Patent), at 11-14 (underlining in original; italics modified).) The patentee made similar statements regarding the Leshem reference, United States Patent Application Publication No. 2002/0147805:

> Applicant recites an arbitrary object framework that separates a content of a computer application, a form of the computer application, and a functionality of the computer application. It is alleged in the Office action that, where Leshem teaches 'self-contained' documents at paragraph [0051], that it is teaching 'separating' content, form, and functionality, recited by Applicant. However, it is respectfully pointed out that Leshem is teaching a document that includes both content and form instructions, which is a combination, the opposite of a separation. Leshem thus clearly fails to teach or even suggest a separation of content, form, and functionality . . . .

(Dkt. No. 136, Ex. I, 5/1/2009 Amendment and Response ('629 Patent), at 7 (underlining in original; italics omitted).)

> In clear contrast to arbitrary objects, prior art object-oriented systems, such as Leshem, utilize "classic" objects which . . . do not utilize *separate content, form, and functionality objects* . . . .

(Dkt. No. 136, Ex. K, 12/30/2009 Amendment and Response ('629 Patent), at 15 (emphasis modified).)

The examiner's reasons for allowance as to the '744 Patent state that the closest prior art references, including Lewandowski, "neither teach or suggest the features of separating content form and functionality of an application and *creating arbitrary objects for each* to generate an application in an arbitrary object framework." (Dkt. No. 136, Ex. G, 8/20/2004 Notice of Allowability ('744 Patent), at 2 (emphasis added).) The examiner's understanding of the claimed invention can be taken into account. *See Am. Hoist & Derrick Co. v. Sowa & Sons, Inc.*, 725 F.2d 1350, 1359 (Fed. Cir. 1984) (noting that patent examiners are "assumed . . . to be familiar from their work with the level of skill in the art"), *abrogated on other grounds, Therasense, Inc. v. Becton, Dickinson & Co.*, 649 F.3d 1276 (Fed. Cir. 2011); *PowerOasis, Inc.*

v. T-Mobile USA, Inc., 522 F.3d 1299, 1304 (Fed. Cir. 2008) (citing *American Hoist*); *Salazar v. Procter & Gamble Co.*, 414 F.3d 1342, 1347 (Fed. Cir. 2005) ("Statements about a claim term made by an Examiner during prosecution of an application may be evidence of how one of skill in the art understood the term at the time the application was filed.").

During reexamination, the patentee explained with regard to the Borland reference, "Borland® Delphi 3 for Windows 95 & Windows NT, User's Guide" (1997), that "[f]orm objects, function objects, and content objects are all 'swapped' (*i.e.*, added or removed) to and from a project, but that is not the same as, *e.g.*, swapping a form object for a content object or a function object." (Dkt. No. 136, Ex. M, 5/31/2012 Amendment ('744 Patent Reexamination), at 13.)

Interwoven considered the specification and the original prosecution history and noted that the patentee "refers to the separation of content, form, and functionality throughout its written description and continuously emphasizes the invention's ability to separate these components during prosecution. The language must be construed for limiting purposes." *Interwoven*, 2011 WL 6936186, at *15. *Interwoven* nonetheless concluded that "within the invention content, form, and functionality, may, not must, be completely separated." *Id.*, at *16. In other words, "although form, content, and functionality may be separated into three distinct entities, they may also be combined within a specific arbitrary object." *Id.*, at *11. *Interwoven* explained that the disclosures in the specification

> merely indicate that the inventions are *capable* of separating form, function, and content, not that such separation is *essential*. In fact, the specification contains language suggesting the possibility of overlap between form, functionality, and content in some instances. For example, in describing the processes necessary to create complex software, the patents explain that "[f]orm includes informative content." ['744 Patent at 1:18.] Furthermore, throughout the patent, reference is made to arbitrary objects which *could* combine content and form, form and functionality, or content and functionality. The specification, therefore, signifies

> that the patents create a new technology which could separate form, function, and content into three distinct components, or could merely separate out one from the other two components. The point of the invention, therefore, is not that all three components are *always* distinct, but rather that the user of the arbitrary objects can choose to separate out either form, function, or content individually.
>
> * * *
>
> [T]he innovation of the patents is the ability to separate form from function from content, but such separation is not required in every instance.

Id., at *6. *Interwoven* also considered the patentee's statements during prosecution regarding the Johnson and Leshem references:

> [The patentee] repeatedly underscores the invention's ability to separate form, content, and functionality. Nowhere in the prosecution history, however, does [the patentee] suggest that such separation is necessary for purposes of its patents. *There is no indication that form, content, and functionality must be distinct, rather, the components may be distinct. What distinguishes this patent from prior art is not the absolute separation of form, content, and functionality, but the user's ability to separate out either one, two, or all three of these components; prior art only permitted a user to separate out one or two at a time.* This reading of the prosecution history is consistent with the remainder of the intrinsic record. The patent language emphasizes that the significance of the invention is the flexibility of being able to separate out the three components, not the necessity of so doing under every circumstance.

Id., at *8 (emphasis added). Finally, *Interwoven* noted the ability to access and modify content, form, and functionality independently. *Id.*, at *16 (citing '744 Patent at 2:9-14). *Interwoven* concluded by construing the "that separates . . ." term to mean "the ability independently to modify and access Content, Form, and Function Objects." *Id.*, at *16.

Defendants bear the burden of proving indefiniteness by clear and convincing evidence. *Halliburton*, 514 F.3d at 1249-50. Further, an indefiniteness finding is inappropriate where a "narrowing construction can properly be adopted." *Exxon*, 265 F.3d at 1375 (citations and internal quotation marks omitted). Here, a "narrowing construction" for the "that separates . . ." term is readily apparent and was adopted in *Interwoven*, as discussed above. That finding in

Interwoven, although subject to independent review by this Court, is entitled to "reasoned deference." *Maurice Mitchell Innovations*, 2006 WL 1751779, at *4.

On balance, in light of the burden of proving indefiniteness by clear and convincing evidence, the ready availability of a narrowing construction, the reasoned deference generally afforded prior constructions, and the intrinsic evidence that refers to separating form, content, and functionality at the system level rather than at the object level (*see* '744 Patent at 2:9-14, 3:14-22 & Claims 1, 54 & 56; *see also* '629 Patent at Claims 1 and 21), the Court reaches the same conclusions here as were reached by *Interwoven*, quoted above. The Court only modifies the *Interwoven* construction so as to improve clarity and to better comport with the above-quoted analysis from *Interwoven*. Defendants' proposal of requiring "content-only, form-only, and functionality-only arbitrary objects" is hereby expressly rejected.

Thus, the Court hereby construes **"that separates a content of said computer application, a form of said computer application and a functionality of said computer application"** to mean **"that enables independently modifying and accessing content, form, and function."**

As Defendants acknowledged at the September 4, 2013 hearing, such a construction of the "that separates . . ." term is *not* inconsistent with "the object being an entity that can have form, content, or functionality or any combination of form, content and functionality." *But see Competitive Techs., Inc. v. Fujitsu Ltd.*, No. 05-1237, 185 F. App'x 958, 965-66 (Fed. Cir. June 15, 2006) ("Because the 'address means' limitation . . . requires ISA structures, and the 'sustain means' limitation of th[e] same claim excludes ISA structures, a person of ordinary skill in the art would be unable to determine the scope of the claims. They are internally inconsistent.

We therefore conclude that the court did not err in holding th[e] claims . . . invalid because of indefiniteness."). Defendants' indefiniteness argument is accordingly hereby expressly rejected.

The Court therefore hereby construes **"the object being an entity that can have form, content, or functionality or any combination of form, content and functionality"** to have its **plain meaning**.

D. "arbitrary object framework"

Plaintiff's Proposed Construction	**Defendants' Proposed Construction**
"a hierarchical structure of arbitrary objects, within which the arbitrary objects by various object types are created by the user based on individual preference managed in an object library, and deployed into a design framework to generate a computer application or deployed into a container page to create a web site"	"a framework that uses only arbitrary objects"

(Dkt. No. 127, Ex. A at 3; *id.*, Ex. B, at 3.) This term appears in Claims 1 and 54-57 of the '744 Patent and Claims 1, 13 and 21 of the '629 Patent. Plaintiff proposes the construction reached by *Interwoven* (which was the construction proposed by Interwoven, Inc. in the *Interwoven* case). *See* 2011 WL 6936186, at *15.

(1) The Parties' Positions

Plaintiff submits that "the claims themselves describe the 'arbitrary object framework' immediately after its appearance in the claims." (Dkt. No. 130, at 21.) Plaintiff argues that Defendants' proposal would improperly limit the term to a preferred embodiment. (*Id.*)

Defendants respond by noting Plaintiff's distinction between "arbitrary objects" and "classic objects" and by arguing that this distinction would be lost if an "arbitrary object framework" could include classic objects. (Dkt. No. 136, at 26.)

Plaintiff replies:

> There is no statement, no express disavowal, no technical reason and no suggestion that the framework contain only arbitrary objects. The statements that the defendants cite refer to arbitrary objects, not the arbitrary object framework. In addition, the defendants argue that if the framework included 'classic objects,' it could not accomplish separation. The record includes nothing to support this allegation. It is technically incorrect.

(Dkt. No. 138, at 9.)

(2) Analysis

The parties dispute whether the "arbitrary object framework" must be limited to including only arbitrary objects. The specification discloses:

> The present invention provides a system and method for using a hierarchical, *arbitrary object framework* for generating software applications. The method separates content, form, and function of the software application so that each can be accessed or modified independently. The method of this invention includes creating arbitrary objects, managing the arbitrary objects in an object library, and deploying the arbitrary objects in a design *framework* for use in computer applications.
>
> * * *
>
> The hierarchical *framework* separates content 10, form 12, and functionality 14 to generate product 16.
>
> * * *
>
> In the method of the present invention, since all that exists within the *framework* is an arbitrary object, the arbitrary object can be swapped for another object that pulls the current piece content in question.

('744 Patent at 3:14-23, 3:32-34 & 4:2-6 (emphasis added).)

> Arbitrary objects may include any combination of application logic and data desired by a developer. Arbitrary objects can include text file pointers, binary file pointers, compiled executable scripts, database queries, shell commands, remote call procedures, global variables and local variables. Arbitrary objects may also include cached data queries and executables. The *arbitrary object framework* allows arbitrary objects to be referenced in a consistent manner regardless of the type of object. Also, the *arbitrary object framework* allows local arbitrary objects to either override global parent arbitrary objects or inherit capabilities and data from the global parent arbitrary object.

> Arbitrary objects can execute any function that can be run or understood by the host computer system so that any underlying functionality of the operating system used by the host system can be defined as an object within the *arbitrary framework*. Legacy data, document objects, CPI [*sic*, CGI] programs, and database queries can all be encapsulated as objects within the *arbitrary framework*. The arbitrary object can be accessed by an arbitrary object name. Arbitrary objects are not tied to their functionality. One arbitrary object can be easily replaced with another arbitrary object of another type.

(*Id.* at 4:20-41 (emphasis added).)

> Many functions are stored within an object library on an *arbitrary object framework* such that those functions can be accessed by name arbitrarily. This is in contrast to a traditional model where the function must be explicitly invoked with all its parameters included. Objects may execute any function that can be run or understood by the host computer system so that any underlying functionality of the host's operating system can be defined as an object within the *framework* of the method of the present invention. The object library can contain legacy data, document objects, CTI [*sic*, CGI] programs, and/or database queries, that can all be encapsulated as objects within a *framework* and accessed from within a design. All that is needed is the name of the function in order to access the function.

(*Id.* at 5:41-54 (emphasis added).)

Interwoven noted, in the course of finding that the term "arbitrary object framework" did not render the claims indefinite, that "[t]his type of framework is presented as the central component for creating, managing, and deploying arbitrary objects." *Interwoven*, 2011 WL 6936186, at *12. For example, the "framework" keeps track of the parameters that each arbitrary object requires so that arbitrary objects can be called by name without parameters. (*See* '744 Patent at 6:2-5 ("The environment space can be available to all objects executed and an object can arbitrarily take advantage of any of the environmental information, depending on the design of the object.").) *Interwoven* concluded that "[u]sing the directions within the specification which explain how an *arbitrary object framework functions to separate, generate, manage, and deploy arbitrary objects*, a person skilled in the art would be able to comprehend the scope of the patent." 2011 WL 6936186, at *15 (emphasis added).

Yet, although the patents-in-suit demonstrate that the "arbitrary object framework" must include arbitrary objects, Defendants have failed to identify any "intentional disclaimer, or disavowal, of claim scope by the inventor" in the specification or any "definitive statements" in the prosecution history that would warrant limiting the "arbitrary object framework" to using only arbitrary objects. *Phillips*, 415 F.3d at 1316; *Omega Eng'g*, 334 F.3d at 1324.

Finally, the parties agreed at the September 4, 2013 hearing that "generate" is synonymous with "create" for purposes of claim construction here.

The Court therefore hereby construes **"arbitrary object framework"** to mean **"a hierarchical structure of arbitrary objects, of various object types, where the arbitrary objects are created by the user, managed in an object library, and deployed into a design framework to create a computer application or deployed into a container page to create a web site."**

E. "deploying said arbitrary objects from said [arbitrary] object library into a design framework to create said computer application"

Plaintiff's Proposed Construction	**Defendants' Proposed Construction**
Plain and ordinary meaning In its opening brief, Plaintiff added an alternative proposal: "moving arbitrary objects into a design framework" (Dkt. No. 130, at 23)	"importing the arbitrary objects from the object library into a design framework and compiling and linking the resulting source code to create a computer application"

(Dkt. No. 127, Ex. A, at 7; *id.*, Ex. B, at 7.) This term appears in Claims 1, 54, and 56 of the '744 Patent and Claims 1 and 21 of the '629 Patent and was not construed by *Interwoven*.

(1) The Parties' Positions

Plaintiff submits that "neither the word 'compiling' nor the word 'linking' appears anywhere in the specification or claims." (Dkt. No. 130, at 23.) Plaintiff argues that Defendants' proposal should be rejected because "neither the process of compiling nor that of linking modules or programs together has anything to do with moving or deploying arbitrary objects." (*Id.*, at 24.) Plaintiff explains that "[c]ompiling and linking change the form of a program, module or object; deploying simply moves it and/or places it into service." (*Id.*) Plaintiff also submits that Defendants' proposed phrase "resulting source code" does not appear in the specification or the claims and "is a slang term used by programmers to mean a program that is in a source language and in a form that has not yet been compiled." (*Id.*)

Defendants respond that the parties agree upon the concepts of importing, compiling, and linking, but "Plaintiff's proposed construction ignores the 'creating' phrase," which "necessarily involves compiling and linking." (Dkt. No. 136, at 25.) Defendants cite extrinsic dictionary definitions of "computer program," "compile," and "link." (*Id.*, Ex. V, *Microsoft Computer Dictionary* 115, 120, 312 (5th ed. 2002).)

Plaintiff replies that the specification "use[s] the word 'deploying' in the commonly accepted (among those skilled in the art) sense of moving" and that "[l]inking and compiling is not even [disclosed in the specification as] an example, and thus does not have any relevance in a discussion of deploying." (Dkt. No. 138, at 10.)

At the September 4, 2013 hearing, Defendants acknowledged that they are relying upon extrinsic evidence, and not any intrinsic evidence, for their proposal of including "compiling" and "linking" in the construction.

(2) Analysis

Claim 1 of the '629 Patent is representative and recites (as amended by the reexamination certificate; emphasis added):

> 1. A system for generating a computer application on a host system in an arbitrary object framework that separates a content of said computer application, a form of said computer application, and a functionality of said computer application, said system including a computer comprising a processor and a memory operably coupled to said processor, said memory being configured for storing a computer program executable by said processor, said computer program comprising:
>
> a first set of executable instructions for creating arbitrary objects with corresponding arbitrary names of content objects used in generating said content of said computer application, form objects used in defining said form of said computer application, and function objects used in executing said functionality of said computer application each arbitrary object being separate from each other arbitrary object, said arbitrary objects being objects that can be *created independently by individual preference*, that are interchangeable, and that may be, but need not be, accessed solely by name, the object being an entity that can have form, content, or functionality or any combination of form, content and functionality;
>
> a second set of executable instructions for managing said arbitrary objects in an arbitrary object library; and
>
> a third set of executable instructions for *deploying said arbitrary objects from said arbitrary object library into a design framework to create said computer application.*

The specification refers to "deploying" arbitrary objects in a design framework or into a page:

> The method of this invention includes creating arbitrary objects, managing the arbitrary objects in an object library, and *deploying* the arbitrary objects in a design framework for use in computer applications.
>
> * * *
>
> At step 24, objects can be *deployed* from the object library into a design framework to create the software application.
>
> * * *
>
> Arbitrary objects can be *deployed* from the object library into a container page to generate the web site.

* * *

> The present invention provides still another technical advantage in that it allows for personalization. Personalization enables companies that want to take advantage of a customer profile to look at the customer's preferences or histories and *deploy* information to the web site specific to the customer.

('744 Patent at 3:19-23, 3:56-58, 4:55-56 & 6:56-61 (emphasis added).) The patents-in-suit do not refer to compiling, aside from references to "compiled executables." (*Id.* at 3:44-45 & 4:23.) Likewise, the patents-in-suit do not refer to resulting source code, importing, or linking. Defendants' proposed construction, which relies upon extrinsic dictionary definitions, lacks sufficient support to be included in the Court's construction. *See Phillips*, 415 F.3d at 1321 (noting that "heavy reliance on the dictionary divorced from the intrinsic evidence risks transforming the meaning of the claim term to the artisan into the meaning of the term in the abstract, out of its particular context, which is the specification").

Defendants' proposed construction, including Defendants' proposal of "compiling and linking," is therefore hereby expressly rejected.

No further construction is necessary. *See U.S. Surgical Corp. v. Ethicon, Inc.*, 103 F.3d 1554, 1568 (Fed. Cir. 1997) ("Claim construction is a matter of resolution of disputed meanings and technical scope, to clarify and when necessary to explain what the patentee covered by the claims, for use in the determination of infringement. It is not an obligatory exercise in redundancy."); *see also O2 Micro Int'l Ltd. v. Beyond Innovation Tech. Co.*, 521 F.3d 1351, 1362 (Fed. Cir. 2008) ("[D]istrict courts are not (and should not be) required to construe every limitation present in a patent's asserted claims."); *Finjan, Inc. v. Secure Computing Corp.*, 626 F.3d 1197, 1207 (Fed. Cir. 2010) ("Unlike *O2 Micro*, where the court failed to resolve the parties' quarrel, the district court rejected Defendants' construction.").

The Court therefore hereby construes **"deploying said arbitrary objects from said [arbitrary] object library into a design framework to create said computer application"** to have its **plain meaning**. To be clear, the Court also rejects Plaintiff's alternative construction, i.e., "moving arbitrary objects into a design framework," because such construction disregards the "to create" phrase in the claim term.

F. "deploying arbitrary objects locally"

Plaintiff's Proposed Construction	**Defendants' Proposed Construction**
Plain and ordinary meaning	Indefinite

(Dkt. No. 127, Ex. A at 8; *id.*, Ex. B, at 9.) This term appears in Claims 17 and 39 of the '744 Patent and Claims 4, 15, and 24 of the '629 Patent and was not construed by *Interwoven*.

(1) The Parties' Positions

Plaintiff argues that "the terms 'local' and its opposite 'global' are very well-known and commonly used in the art of computer programming" in accordance with a "concept . . . known in the art as 'scope.'" (Dkt. No. 130, at 25.) Plaintiff explains that "[a] global variable, for example, can be seen and accessed by all programs or objects in an application, whereas a local variable can only be seen or accessed by the object or function that contains it." (*Id.*) Plaintiff concludes that "since the term 'local' (and, hence, the adverbial form, 'locally') is a well-known term in the art of computer programming, the term 'deploying arbitrary objects locally' is not indefinite and should have its plain and ordinary meaning in the art." (*Id.* (citing Ex. K, *Microsoft Computer Dictionary* 315 (5th ed. 2002)).)

Defendants respond that although the specification discloses a "content management system," "host computer system," "operating system," and "object oriented development systems," the disputed term "is insolubly ambiguous because one skilled in the art is left

guessing which 'system' the claim refers to." (Dkt. No. 136, at 29-30.) Defendants also argue that "[Plaintiff's] paraphrased definition of 'local' from the Microsoft Computer Dictionary is not helpful because it defines 'local' in terms of a variable within a program, not an object within a system. [Plaintiff] cannot rely on a technical dictionary to define 'local' that clearly runs counter to the specification of the patents-in-suit as a basis for the belief that this term is a 'well-known term in the art of computer programming.'" (*Id.*, at 30 (citations omitted).) Defendants conclude that "[w]ith no clear definition in the intrinsic record and no support for a plain and ordinary meaning in the art, this term is indefinite." (*Id.*)

Plaintiff's reply brief addresses this term together with the "deploying said arbitrary objects . . ." term, discussed above. (*See* Dkt. No. 138, at 9-10.)

(2) Analysis

Claim 24 of the '629 Patent is representative and recites (emphasis added): "24. The system of claim 21, wherein the third set of executable instructions are [*sic*, is] for *deploying arbitrary objects locally*."

Although the specification discusses several different systems, e.g., "content management system" ('744 Patent at 3:61-62), "host computer system" (*id.* at 4:34), "operating system" (*id.* at 4:35), and "object oriented development systems" (*id.* at 5:63), the specification explains that "[o]bjects may be deployed *globally* across an entire system or *locally* within a specific area or sub-areas of a system." ('744 Patent at 4:13-14 (emphasis added).)

The Court therefore hereby construes **"deploying arbitrary objects locally"** to mean **"moving arbitrary objects within a specific area or sub-areas of a system."**

Consequently, Defendants have failed to "show[] by clear and convincing evidence that a skilled artisan could not discern the boundaries of the claim based on the claim language, the

specification, and the prosecution history, as well as her knowledge of the relevant art area." *Halliburton*, 514 F.3d at 1249-50. Defendants' indefiniteness argument is hereby expressly rejected.

IV. CONCLUSION

The Court adopts the constructions set forth in this opinion for the disputed terms of the patents-in-suit. The parties are ordered that they may not refer, directly or indirectly, to each other's claim construction positions in the presence of the jury. Likewise, the parties are ordered to refrain from mentioning any portion of this opinion, other than the actual definitions adopted by the Court, in the presence of the jury. Any reference to claim construction proceedings is limited to informing the jury of the definitions adopted by the Court.

Within thirty (30) days of the issuance of this Memorandum Opinion and Order, the parties are hereby ORDERED, in good faith, to mediate this case with the mediator agreed upon by the parties. As a part of such mediation, each party shall appear by counsel and by at least one corporate officer possessing sufficient authority and control to unilaterally make binding decisions for the corporation adequate to address any good faith offer or counteroffer of settlement that might arise during such mediation. Failure to do so shall be deemed by the Court as a failure to mediate in good faith and may subject that party to such sanctions as the Court deems appropriate.

So ORDERED and SIGNED this 16th day of September, 2013.

RODNEY GILSTRAP
UNITED STATES DISTRICT JUDGE